COMPUTER SCIENCES AND SOFTWARE ENGINEERING
AUBURN UNVERSITY

COMP 2710
Software Construction

Summer 2013

Lab 1
A Simple Meeting Room Application

Due: June 19, 2013

Points Possible: 100 (25 Design, 75 code)
Due:
Design Portion: **Wednesday, June 12th, 2013 by 11:55 pm** turned in via CANVAS SUBMISSION
Programming Portion: **Wednesday, June 19th, 2013 by 11:55 pm** turned in via CANVAS SUBMISSION

Goals:
To develop some proficiency with basic C++ syntax and semantics.
To organize code into classes.
To gain some familiarity with formal testing.
To write a fun application!

Design Portion:
Create a text or PDF file named "<username>_1w.txt" (for example, mine would read "lim_1w.txt"). Please submit a text (.txt) file or PDF (.pdf), *do not submit MS Word Documents or other proprietary formats.*

Process (steps 1-3 due with initial turn-in, steps 4-5 due with final turn-in:
Each of the following should address the program specification below. Concern yourself more with thinking about the problem than worrying about the specifics of implementation at this stage. For the first turn-in, you will provide steps 1-3. For the final (electronic) turn-in, you will provide all the steps (including any revisions you may have made). Please note revisions in your final copy! You should provide:

1. **Analysis (10):** Write a few important **use cases**. Remember, these describe how a user interacts with the system (what they do, what the systems does in response, etc.). These need not be long, but they should have enough basic details such that someone unfamiliar with the system can have an understanding of what is happening. They should not include internal technical details that the user is not (and should not) be aware of. Make sure that any special rules/features you plan to add are clearly described in your Analysis section.

2. **Design (10)**:

Your design must include the following:

A. Create a Class Diagram (as in Lab 2). From your use cases, identify likely candidates for software objects. Be sure to include:

1) The name and purpose of the class
2) The member variables and the functions of the class
3) Any other classes that this class depends on or uses
4) Any relevant notes that don't fit into the previous categories

B. Create the data flow diagrams. Show all the entities and processes that comprise the overall system and show how information flows from and to each process.

3. **Testing (5)**: Develop lists of specific test cases:

1) For the system at large. In other words, describe inputs for "nominal" usage. You may need several scenarios. Also, suggest scenarios for abnormal usage and show what your program should do (for example, entering a negative number for a menu might ask the user to try again).
2) For each object. (Later, these tests can be automated easily using a simple driver function in the object).

Remember, test cases are specific, repeatable, and should have a clearly defined result.

4. Implement your solution in C++ (70).

5. **Test Results (5)**: ***After developing your solution***, actually try all of your test cases (both system and unit testing). Actually show the results of your testing (a copy and paste from your program output is fine – don't stress too much about formatting as long as the results are clear). You should have test results for every test case you described. If your system doesn't behave as intended, you should note this.

Program Portion:

One of the most commonly used application in the smart phones, laptops and desktops is the electronic meeting room, messaging or texting application. In this Lab 1, you will analyze, design, implement and test a simple meeting room application. It will have a simple text-based user interface that allows multiple users to send and receive messages to multiple meeting rooms. In this lab project, all the users and message buffers must be implemented in the same address space, i.e. the meeting room message buffers memory space can be shared by all the users.

What is an electronic meeting room?

You hopefully have *some* idea of what an electronic meeting room is. There are lots of different kinds, but for this assignment's purposes, a meeting room is a program (or collection of programs) that does the following:

- There can be lots of different meeting rooms, identified by a meeting room name. In this lab project, you will only need to design and implement only two meeting rooms. The user must be allowed to enter any name for the meeting room.
- People can "connect to" or "enter" a meeting room. In this lab project, you must implement at least three users. The user must be allowed to enter any name.

- Each user can send messages to a meeting room to which he/she is connected. Users who are connected to a meeting room can receive all messages sent to the meeting room. *Users must not see messages that were sent before they connected to the meeting room, only those messages that are sent after they have connected to the meeting room.*
- Each user has the option of either displaying just the new messages so far or all messages since they connected to the meeting room.
- Each message must contain the name of the user and the text message.

Message format

For this assignment, design your own meeting room messages using any appropriate format. The main requirements are that the message must contain the user's name and each message can be separated easily from another message. However, each message must be of variable length. If you like, you can use the message format that is used in Homework 1, i.e. use ": " to separate the user name from the message text and use "\n" to separate the messages. The purpose for this standard message format is that everyone's meeting room should end up being compatible. That is, every user should be able to work with every meeting room.

The user interface

Write a menu-based and text-based user interface for interacting with the meeting room system. The main menu has (at least) 9 options to choose from as shown below. *You can abbreviate the menu so that all the 9 options can be shown in only one or two lines.* The user interface will accept the option and all related inputs from the user, perform the operation and then repeatedly shows this menu of options again and prompts for the next option.

- **Create a meeting room**: When selected, the program will then prompt for the name of the meeting room and then associate that name with one of the meeting rooms.
- **Create a new user:** When selected, the program will then prompt for the name of the user and associate the name with one of the user objects. When a user is created, the *current user* is now that user so that he/she can connect or enter a meeting room. The program must prompt a welcome banner, e.g. saying "Welcome, Joe!" , which should be centered on the screen and surrounded by a box.
- **Connect to a Meeting room:** When selected, the program will prompt the name of the meeting room. Once connected, the connection will be established for the duration of the meeting room application. If it is a new connection, it will create a new connection to the named meeting room. If there is already a connection to that meeting room, then the existing connection is used. The program should display a banner indicating the current user and the current meeting room, e.g. "Joe is now in the Alpha Meeting room!", which should be centered on the screen and surrounded by a box.
- **Send a message**: When selected, the program will prompt for the message and send message to the *current meeting room.*

- **Display new messages**: When selected, the program will display new messages if they exist. New messages are those that have not already been previously received by that user. If there is no new message, it will display "No new message".
- **Display all messages**: When selected, the program will display all messages from the *current meeting room*, starting from the time the user first connected to the meeting room. If there is no message, it will display "No new message".
- **Switch to a different user**: When selected, the program will prompt for the user name, e.g. Barry, and then switch current user to Barry. The program must check to make sure that a user with that name has already been created, otherwise it should display an error and repeatedly ask for the user name that exists. (Your program must not exit or terminate when this naming error occur.) When switched to a different user, the program will connect that user to the meeting room that he/she was previously connected to. The program must prompt a banner indicating that, e.g. saying "Barry is now in the Beta meeting room!" , which should be centered on the screen and surrounded by a box.
- **Switch to a different meeting room**: When selected, the program will prompt for the meeting room name. The program must first check that the meeting room name exists. If not, the *current meeting room* is set to NULL, and the program displays an error message. It then checks that if the current user is connected to the meeting room. If he/she is already connected to the meeting room, then the *current meeting room* is set to this meeting room. Subsequent messages are sent and received from this meeting room. The program should display a banner indicating the current user and the current meeting room, e.g. "Mary is now in the Beta meeting room!", which should be centered on the screen and surrounded by a box.
- **Quit the meeting room application:** When selected, the program will exit the program.

The user interface must check for correct input value from the users. If there is any error, e.g. incorrect name of the meeting room when switching to a different meeting room, then the program must display the error and continue to prompt for the correct input. You program must not exit or terminate when there is an incorrect input value.

The name of your program must be called <username>_1.cpp (for example, mine would read "lim_1.cpp"

Use comments to provide a heading at the top of your code containing your name, Auburn Userid, and filename. You will lose points if you do not use the specific program file name, or do not have a comment block on **EVERY** program you hand in.

Your program's output need not exactly match the style of the sample output (see the end of this file for one example of sample output).

Important Notes:
You must use an object-oriented programming strategy in order to design and implement this meeting room application (in other words, you will need write class definitions and

use those classes, you can't just throw everything in main()). A well-done implementation will produce a number of robust classes, many of which may be useful for future programs in this course and beyond. Remember good design practices discussed in class:

a) A class should do one thing, and do it well
b) Classes should NOT be highly coupled
c) Classes should be highly cohesive

Some potential classes:

1. A ***Menu*** class which handles basic user screw-ups (choosing an option out of bounds).
2. A ***System*** class which instantiates the other objects that must be initialized.
3. A ***User*** class which maintains information on the user name and the current meeting room that it he/she is connected to.
4. A ***MeetingRoom*** class that maintains the messages and allows messages to be sent (stored) into its buffer and received (retrieved) from the buffer.
5. A ***Connector*** class that maintains the connection between a user and a meeting room, e.g. it keeps track of the messages that have been retrieved.

You should at a very minimum have classes to handle menus, users and meeting rooms. Use your judgement to define additional classes or not use some of the classes above.

- You should follow standard commenting guidelines.

- You DO NOT need any graphical user interface for this simple, text-based application. If you want to implement a visualization of some sort, then that is extra credit.

Error-Checking:

You should provide enough error-checking that a moderately informed user will not crash your program. This should be discovered through your unit-testing. Your prompts should still inform the user of what is expected of them, even if you have error-checking in place.

Please submit your program through the Canvas system online. If for some disastrous reason Canvas goes down, instead e-mail your submission to TA – Sankari Anupindi – at sza0033@tigermail.auburn.edu. Canvas going down is not an excuse for turning in your work late.

You should submit the two files in digital format. No hardcopy is required for the final submission:

<username>_1.cpp
<username>_1p.txt (script of sample execution, especially the results of testing)

Your test results must show posting and display of messages from at least three users in at least two meeting rooms.

Sample Usage:

```
       ===========================================================
       |        Welcome to the Meeting Room Application!         |
       ===========================================================

   1) Create a meeting room
   2) Create a new user
   3) Connect to a Meeting room
   4) Send a message
   5) Display new messages
   6) Display all messages
   7) Switch to a different user
   8) Switch to a different meeting room
   9) Quit the meeting room application


       Please choose an option: 2

       Please enter user name: Mary

       ===========================================================
       |                     Welcome, Mary!                      |
       ===========================================================

   1) Create a meeting room
   2) Create a new user
   3) Connect to a Meeting room
   4) Send a message
   5) Display new messages
   6) Display all messages
   7) Switch to a different user
   8) Switch to a different meeting room
   9) Quit the meeting room application

       Please choose an option: 2

       Please enter user name: Joe


       ===========================================================
       |                     Welcome, Joe!                       |
       ===========================================================
   1) Create a meeting room
   2) Create a new user
   3) Connect to a Meeting room
   4) Send a message
   5) Display new messages
   6) Display all messages
   7) Switch to a different user
   8) Switch to a different meeting room
   9) Quit the meeting room application

       Please choose an option: 2

       Please enter user name: Bill

       ==========================================================
       |                    Welcome, Bill!                      |
       ==========================================================

   1) Create a meeting room
   2) Create a new user
   3) Connect to a meeting room
   4) Send a message
   5) Display new messages
   6) Display all messages
   7) Switch to a different user
   8) Switch to a different meeting room
   9) Quit the meeting room application

       Please choose an option: 1

       Please enter meeting room name: Alpha

       ==========================================================
       |              Alpha Meeting Room created                |
       ==========================================================

   1) Create a meeting room
   2) Create a new user
   3) Connect to a meeting room
   4) Send a message
   5) Display new messages
   6) Display all messages
   7) Switch to a different user
   8) Switch to a different meeting room
   9) Quit the meeting room application

       Please choose an option: 3

       Which meeting room to connect: Alpha

       ==========================================================
       |          Bill is now in the Alpha meeting room!        |
       ==========================================================

   1) Create a meeting room
   2) Create a new user
```

```
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 7

     Please enter user name: Mary

     ===========================================================
     |                      Welcome, Mary!                     |
     ===========================================================

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 3

      Which meeting room to connect?: Alpha

     ===========================================================
     |          Mary is now in the Alpha meeting room!         |
     ===========================================================

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 4

      Enter message: Hello there! Anyone home?

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application
```

```
     Please choose an option: **7**

     Please enter user name: **Bill**

     ===========================================================
     |             Bill is in the Alpha meeting room!          |
     ===========================================================

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 5

     Mary: Hello there! Anyone home?


1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 4

      Enter message: Hi Mary, How are you?


1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 4

      Enter message: Going to the game?


1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
```

```
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 7

     Please enter user name: Mary

     ===========================================================
     |          Mary is in the Alpha meeting room!             |
     ===========================================================

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 5

     Bill: Hi Mary, How are you?
     Bill: Going to the game?


1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 4

     Enter message: Sure, I'll see you at the game. Bye!

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 7
     Please enter user name: Bill

     ===========================================================
     |            Bill is in the Alpha meeting room!           |
     ===========================================================

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 5

     Mary: Sure, I'll see you at the game. Bye!

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 6

     Mary: Hello there! Anyone home?
     Bill: Hi Mary, How are you?
     Bill: Going to the game?
     Mary: Sure, I'll see you at the game. Bye!

1) Create a meeting room
2) Create a new user
3) Connect to a meeting room
4) Send a message
5) Display new messages
6) Display all messages
7) Switch to a different user
8) Switch to a different meeting room
9) Quit the meeting room application

     Please choose an option: 9

     ===========================================================
     |      Thank you for using the meeting room application   |
     ===========================================================
```